

WHAT IS CLAIMED IS:

1. An apparatus for subdividing input data associated with a software program and processing each subdivided input data on one or more processing elements, comprising:
  - an initiating program;
  - one or more processing programs; and
  - a wrapper that intercepts a call to the initiating program and operable to subdivide input parameters into one or more job quanta, wherein each job quantum is submitted for execution to a separate processing program selected from the one or more processing programs residing on a separate processing element.
2. The apparatus of claim 1, wherein the wrapper assembles one or more output data from each processing program to form a single results data.
3. The apparatus of claim 1, wherein each job quantum is provided to a separate job scheduler residing on each of the processing elements, each scheduler manages the execution of the processing program executing on the processing element.
4. The apparatus of claim 1, further comprising:
  - one or more additional wrappers, each additional wrapper residing on a single processing element and is operable to intercept the job quantum submitted to the processing program residing on the processing element.
5. The apparatus of claim 1, wherein the initiating program and each of the processing programs perform one or more operations that are substantially identical.

6. The apparatus of claim 5, wherein the operations are bioinformatic calculations.
7. The apparatus of claim 1, wherein the initiating program and each of the processing programs are non-threaded.
8. The apparatus of claim 7, wherein the initiating program and each of the processing programs execute substantially in parallel.
9. The apparatus of claim 1, wherein at least one of the processing elements resides in a disparate processing environment from the initiating program.
10. The apparatus of claim 1, wherein the input parameters are normalized prior to being subdivided into the job quanta.
11. A method of processing a non-threaded set of executable instructions, comprising:
  - receiving input data associated with a call to a first non-threaded set of executable instructions;
  - parsing the input data into a plurality of job quanta, each job quantum operable to be independently processed by the first non-threaded set of executable instructions; and
  - submitting at least one job quantum for execution to a second non-threaded set of executable instructions that is substantially identical to the first non-threaded set of executable instructions, wherein the second set of executable instructions resides on one or more different processing elements from the first non-threaded set of executable instructions.

12. The method of claim 11, further comprising:  
assembling an output data associated with the results of the execution of the second non-threaded set of executable instructions for a presentation.
13. The method of claim 11, further comprising:  
submitting at least one job quantum for execution to the first non-threaded set of executable instructions.
14. The method of claim 13, wherein the executions occur substantially in parallel.
15. A job quanta data structure, comprising:  
a first data;  
a second data wherein the first and second data are operable to be delineated and independently submitted as input parameter data used for execution by separate non-threaded sets of executable instructions and processed substantially in parallel.
16. The job quanta of claim 15, wherein the first and second data are delineated using extensible markup language.
17. The job quanta of claim 15, wherein the first and second data are initially submitted as input parameter data to a single non-threaded set of executable instructions.
18. A system for performing parallel processing on a call to execute a software program, comprising:  
means for intercepting a call to the software program;

means for dividing a set of input data into a plurality of job quanta including a first job quantum and a second job quantum;

means for submitting the first job quantum to the software program and for submitting the second job quantum to a separate software program; and

means for executing the software program and the separate software programs substantially in parallel.

19. The system of claim 18, further comprising:

means for assembling output data associated with the execution of the software program and at least one of the separate software programs into a presentation data.

20. The system of claim 19, further comprising:

means for trapping and reporting error conditions generated by the execution of the software program and at least one of the separate software programs.

21. A method of processing a software program, comprising:

receiving input data associated with a call to the software program;

parsing the input data into a plurality of job quanta, each job quantum operable to be independently processed by the software program; and

submitting at least one job quantum for execution to a replica software program that is substantially identical to the software program, wherein the replica software program resides on one or more different processing elements from the software program.

22. An information handling system, comprising:

a network;

a plurality of processing elements;

memory operatively coupled to the processing elements; and

means for wrapping a call to an application program by dividing input data among the processing elements for execution according to the application program and recombining output data from the processing elements.